Claim(s)

[0040] Having thus described the preferred embodiments, the invention is now claimed to be:

1. A method of verifying that a CDMA code allocator maintains mutual orthogonality between all concurrently busy codes, said method comprising:

(a) identifying a code being allocated by the allocator;

(b) determining if the identified code is busy;

(c) determining if any ancestral parent of the identified code is busy; and,

(d) determining if any descendant of the identified code is busy;

wherein, if at least one of: the identified code; one of the identified code's ancestral parents; or, one of the identified code's descendants; is determined to be busy, then an error in allocator operation is indicated.

2. The method according to claim 1, further comprising:

stimulating the allocator with an artificial call generator; and,

monitoring an output of the allocator to identify the codes being allocated by the allocator.

3. The method according to claim 1, wherein the allocator is simulated.

4. The method according to claim 1, wherein the determinations of steps (b), (c) and (d) are made by accessing a storage device in which current code states are maintained.

5. The method according to claim 4, wherein the storage device comprises a LUT arranged as a binary tree of codes in which each parent code is a common node for its two children codes.

6. The method according to claim 1, further comprising:

repeating steps (a) through (d) each time a code is allocated by the allocator; and,

saving a historical record of the allocator's operation with respect to maintaining mutual orthogonality between all concurrently busy codes.

7. The method according to claim 6, further comprising:

designating code states such that an otherwise idle code is designated as busy when the code is allocated, and an otherwise busy code is designated as idle when the code is de-allocated.

8. The method according to claim 6, wherein the designations of code states are maintained in a storage device which is accessed to make the determinations of steps (b), (c) and (d).

9. The method according to claim 8, wherein the storage device comprises a LUT arranged as a binary tree of codes in which each parent code is a common node for its two children codes.

10. An allocator testing system for testing a Walsh code allocator to verify that its operation maintains mutual orthogonality between all concurrently busy Walsh codes, said allocator testing system comprising:

a call generator which drives an allocator being tested, said call generator providing an input of the allocator with a pattern of channel openings and closings in response to which the allocator outputs Walsh code allocations; and,

a verification module arranged to receive the allocator outputs, said verification module determining for each Walsh code allocation whether or not it would result in at least two non-orthogonal Walsh codes being concurrently busy.

11. The allocator testing system according to claim 10, further comprising:

a storage device which is accessed by the verification module to determine whether or not a Walsh code allocation would result in at least two non-orthogonal Walsh codes being concurrently busy, said storages device maintaining current states of Walsh codes.

12. The allocator testing system according to claim 11, wherein the storage device comprises a LUT arranged as a binary tree of Walsh codes in which each parent Walsh code is a common node for its two children Walsh codes.

13. The allocator testing system according to claim 10, wherein the verification module determines whether or not an allocated Wash code is busy, whether or not an ancestral parent of an allocated Walsh code is busy, and whether or not a descendant of an allocated Walsh code is busy, such that if at least one of the foregoing is determined to be busy, then an error in the allocator's operation is indicated.

14. The allocator testing system according to claim 10, wherein the call generator is an artificial call generator.

15. The allocator testing system according to claim 10, wherein the allocator being tested is a simulated allocator.

16. The allocator testing system according to claim 10, further comprising:

a storage device in which current states of Walsh codes are maintained, wherein the verification module accesses the storage device to determine whether or not a Wash code being allocated is busy, whether or not an ancestral parent of a Walsh code being allocated is busy, and whether or not a descendant of a Walsh code being allocated is busy, such that if at least one of the foregoing is determined to be busy, then an error in the allocator's operation is indicated.